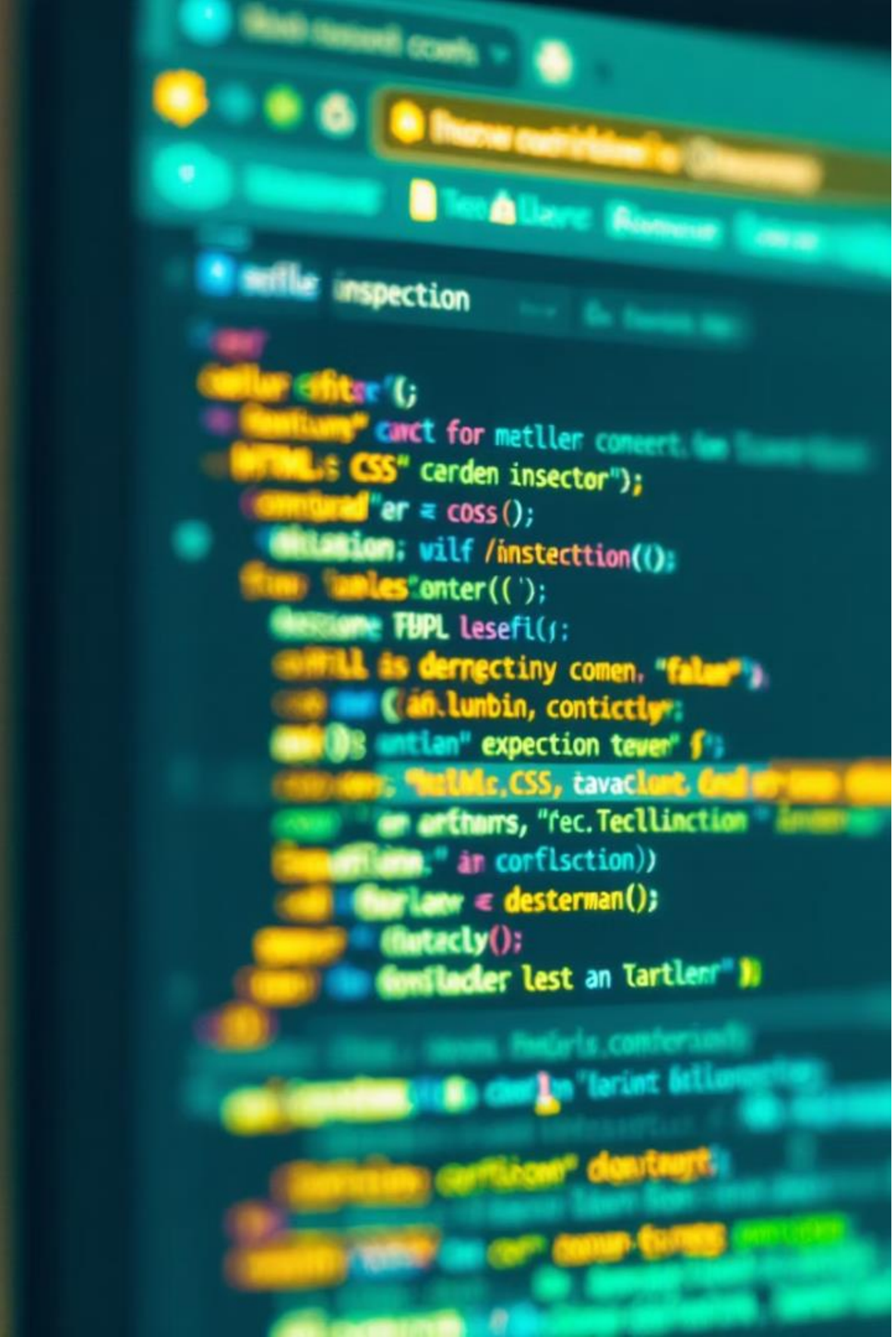
클라이언트 코드 보안을 이해하기

- 사용자가 직접 조작 가능한 클라이언트 코드는 보안상 최대 취약지
- HTML, JS, CSS, 쿠키, 패킷, 메모리까지 모두 포함
- 전체 보안 사고의 70%가 클라이언트 코드 이해 부족에서 시작



클라이언트 코드란?

사용자 관점

브라우저에서 직접 실행 가능한 모든 코드입니다.

포함 범위

HTML, JavaScript, CSS, 쿠키, 메모리, 네트워크 패킷까지 포함합니다.

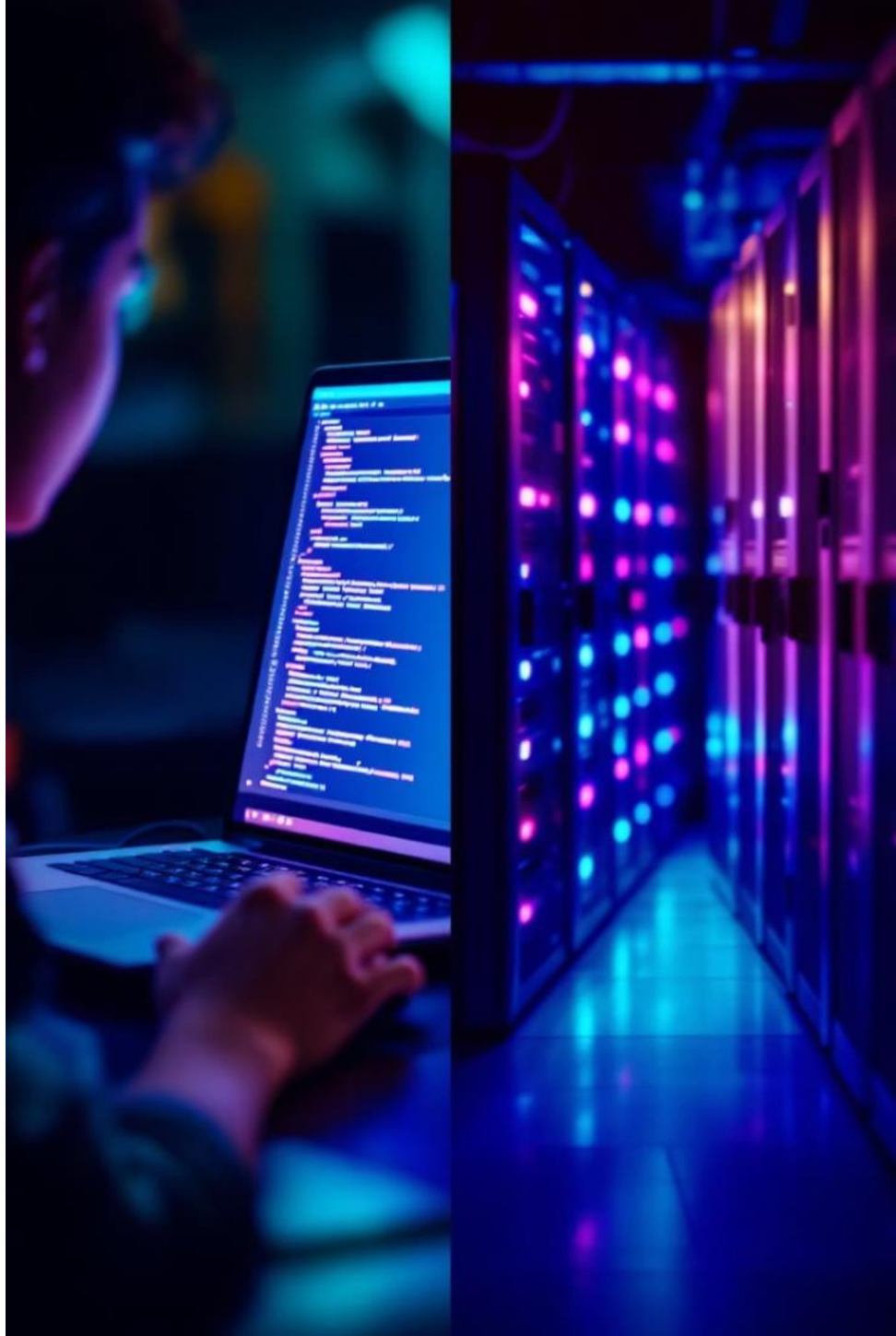
실제 위협

보안 사고 70%는 클라이언트 코드에 대한 무지에서 시작됩니다.

클라이언트 vs 서버 코드

항목	클라이언트	서버
실행 위치	브라우저/기기	서버 내부
접근성	누구나 볼 수 있음	비공개
조작 가능성	매우 높음	불가능

클라이언트 코드의 신뢰는 금물입니다.
모든 입력은 의심하고 서버에서 검증해야 합니다.



웹의 발전과 클라이언트 코드의 변화

1

1세대

정적 HTML, 단순 문서 뷰어

2

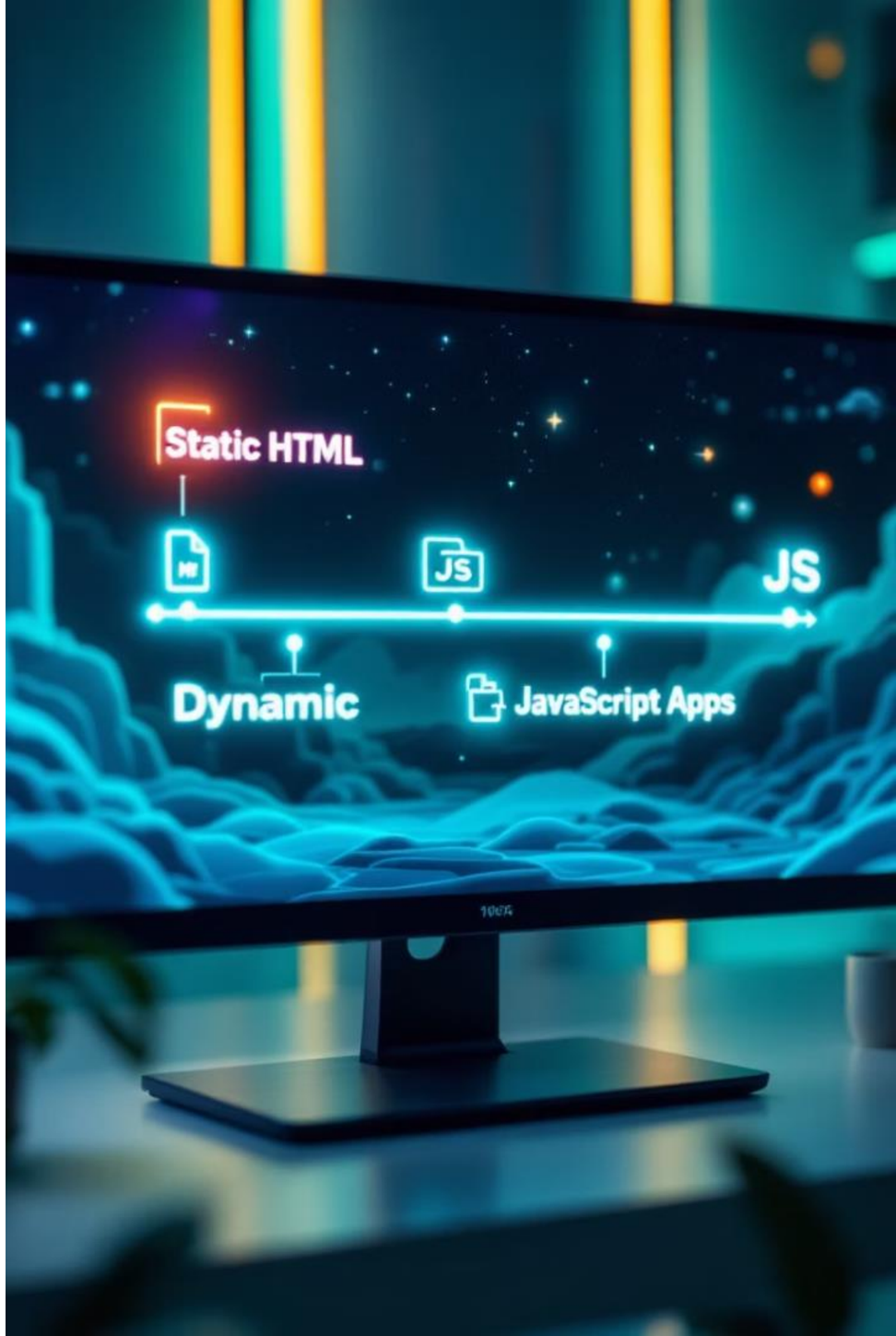
2세대

CSS, JavaScript, 폼 활용한 동적 웹

3

3세대

AJAX 비동기 통신, API 활용, 복잡한 클라이언트 코드



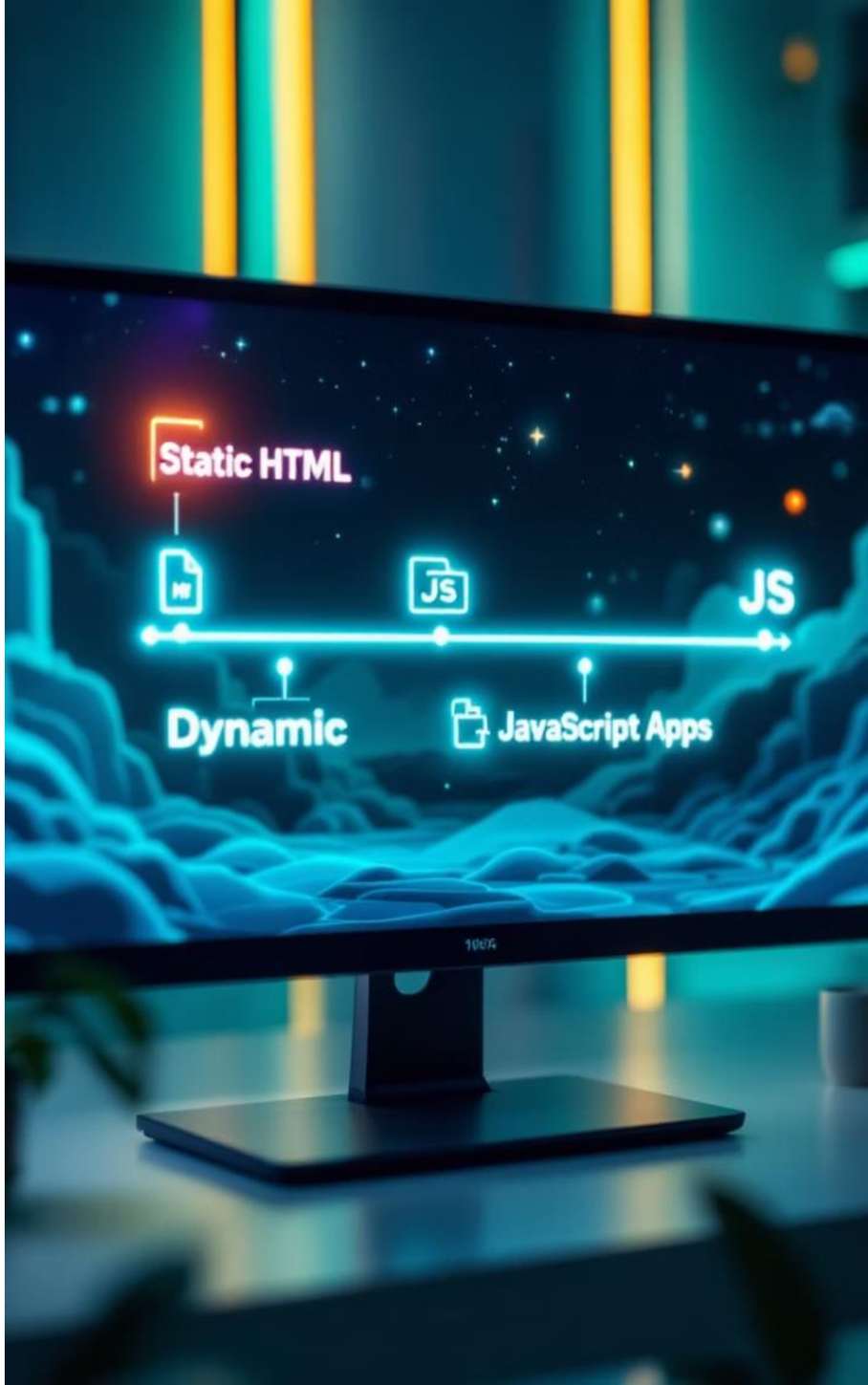
HTTP 메시지의 구조와 흐름 이해

Request 구조

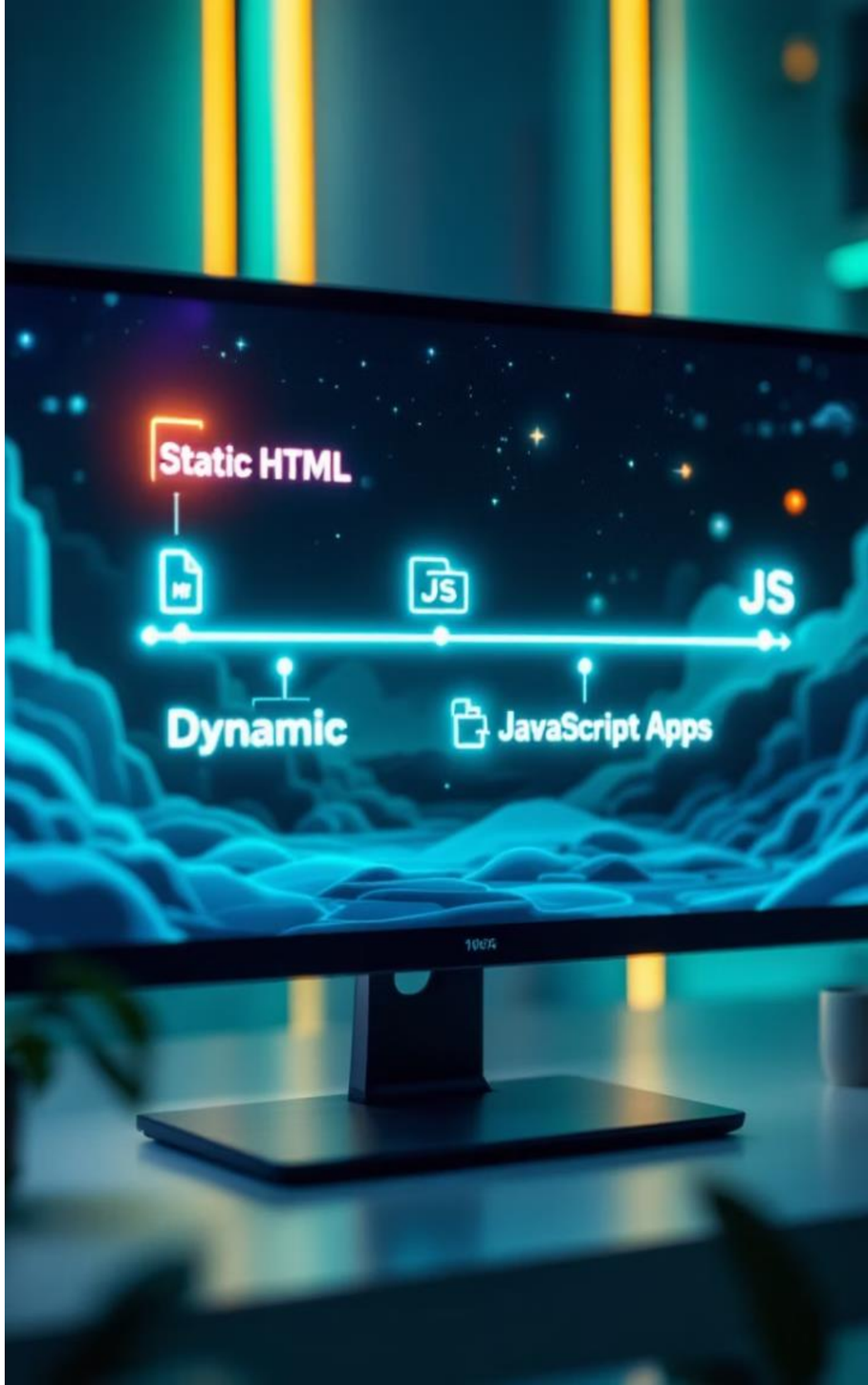
- Request Line: 요청 주소, 메서드
- Headers: User-Agent, Cookie 등
- Body: 폼 데이터, JSON 등

Response 구조

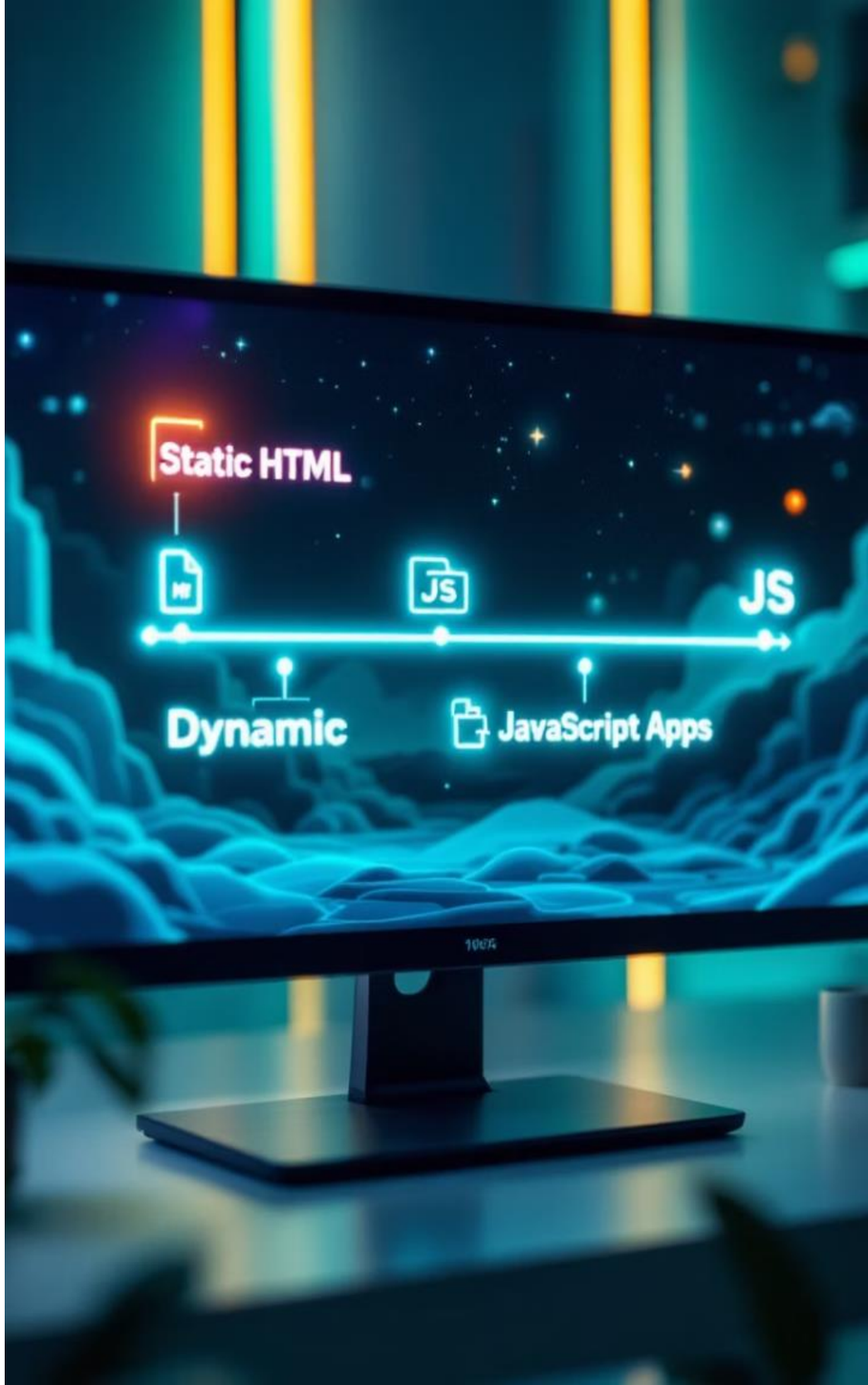
- Status Line: 응답 코드
- Headers: Content-Type 등
- Body: HTML, 이미지, 데이터 등



쿠키 생성 파이썬 예제 코드



피들러로 쿠키 생성 확인
- 설치 및 확인



- HTTP/HTTPS 트래픽을 분석할 수 있는 무료 프록시 도구
- 클라이언트(브라우저, 앱)와 서버 간의 **요청과 응답을 실시간으로 가로채고 분석**
- 웹 개발, 보안 테스트, API 디버깅 등에 유용하게 사용

Mechanism of Proxy Server

Communication Without Proxy Server

Communication With Proxy Server

프록시의 특징

- 보안 및 프라이버시 보호
- 접근 제어 및 콘텐츠 필터링
- 캐시를 통한 속도 향상

Fiddler download for windows

- www.telerik.com/doiwnload/fididler

Telerik.com
https://www.telerik.com › download › fiddler

Download Fiddler Web Debugging Tool for Free by Telerik

Download and install Fiddler Classic web debugging tool. Watch a quick tutorial to get started.

Download Fiddler Everywhere · Fiddler Classic · License Agreement

Telerik.com
https://www.telerik.com › fiddler › fiddler-classic

Web Debugging Proxy Tool | Fiddler Classic

Fiddler Everywhere is a modern, easy-to-use web debugging proxy tool for Windows, macOS and Linux which captures, logs, monitors and inspects all HTTPS traffic, ...

Download Fiddler · Docs & Support · Fiddler Everywhere · FiddlerCore

Telerik.com
https://www.telerik.com › download › fiddler-everywhere

Download Fiddler Everywhere

Download Fiddler Everywhere, the professionally built and supported web debugging proxy tool for Windows, macOS, and Linux. Free and fully-functional trial.

Download Fiddler Classic

Get started with the free web debugging proxy tool exclusively

How do you plan to use Fiddler Classic? *

Select

Email *

Your email

Country/Territory *

-- Select --

I accept the Fiddler End User License Agreement

Download For Windows*

or

How do you plan to use Fiddler Classic? *

Personal use

Email *

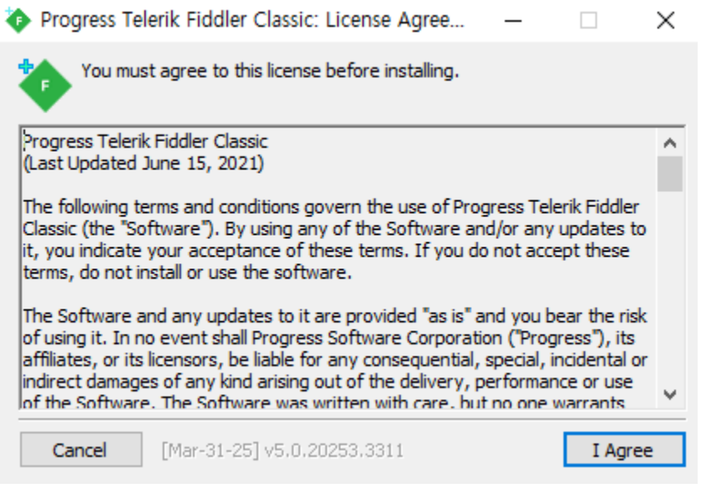
bitbuild@naver.com

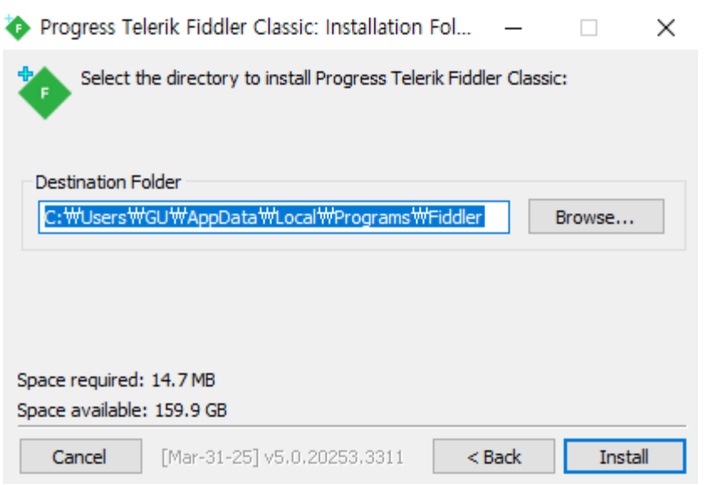
Country/Territory *

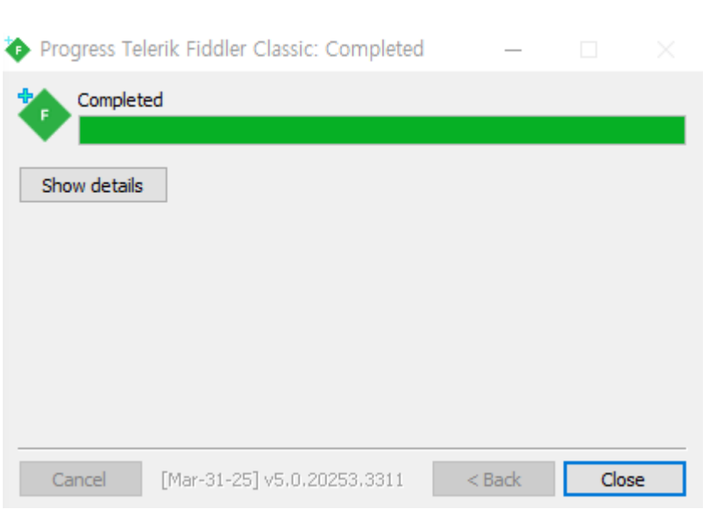
Republic of Korea (South Korea)

By submitting this form, you understand and agree that your personal data will be processed by Progress Software or its Partners as described in our Privacy Policy. You may opt out from marketing communication at any time here or through the opt out option placed in the e-mail communication sent by us or our Partners.

I accept the Fiddler End User License Agreement







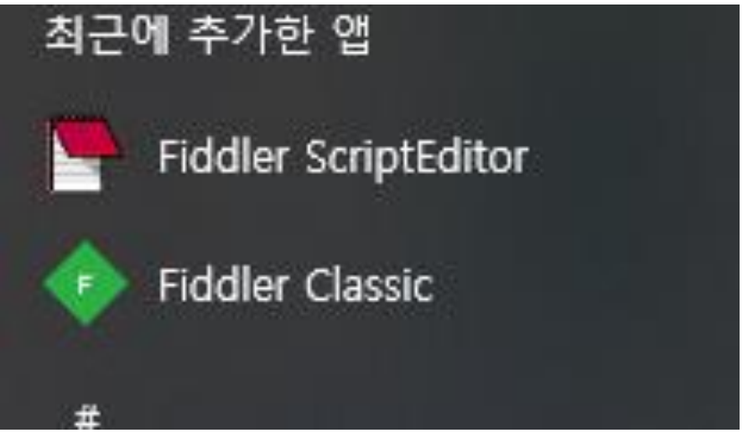
Installation was successful!

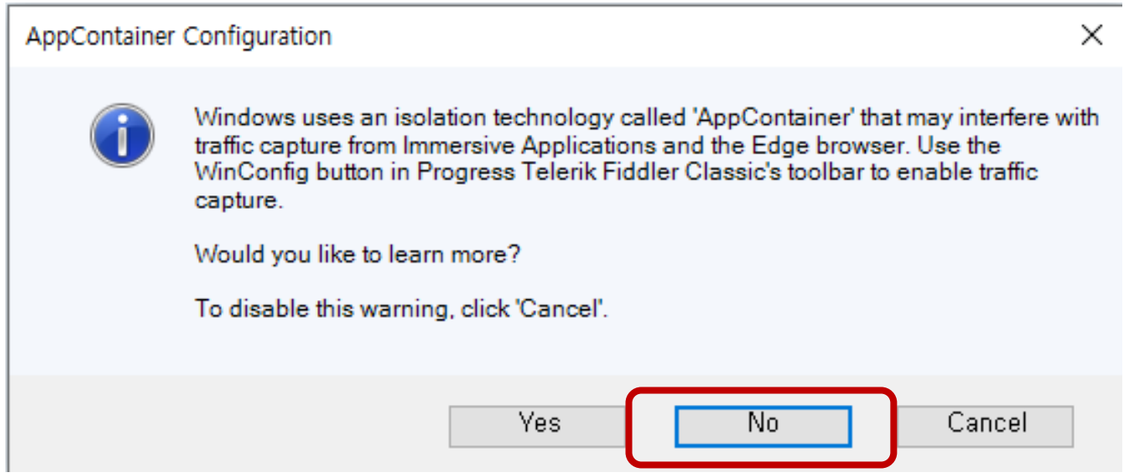
To start Telerik Fiddler Classic

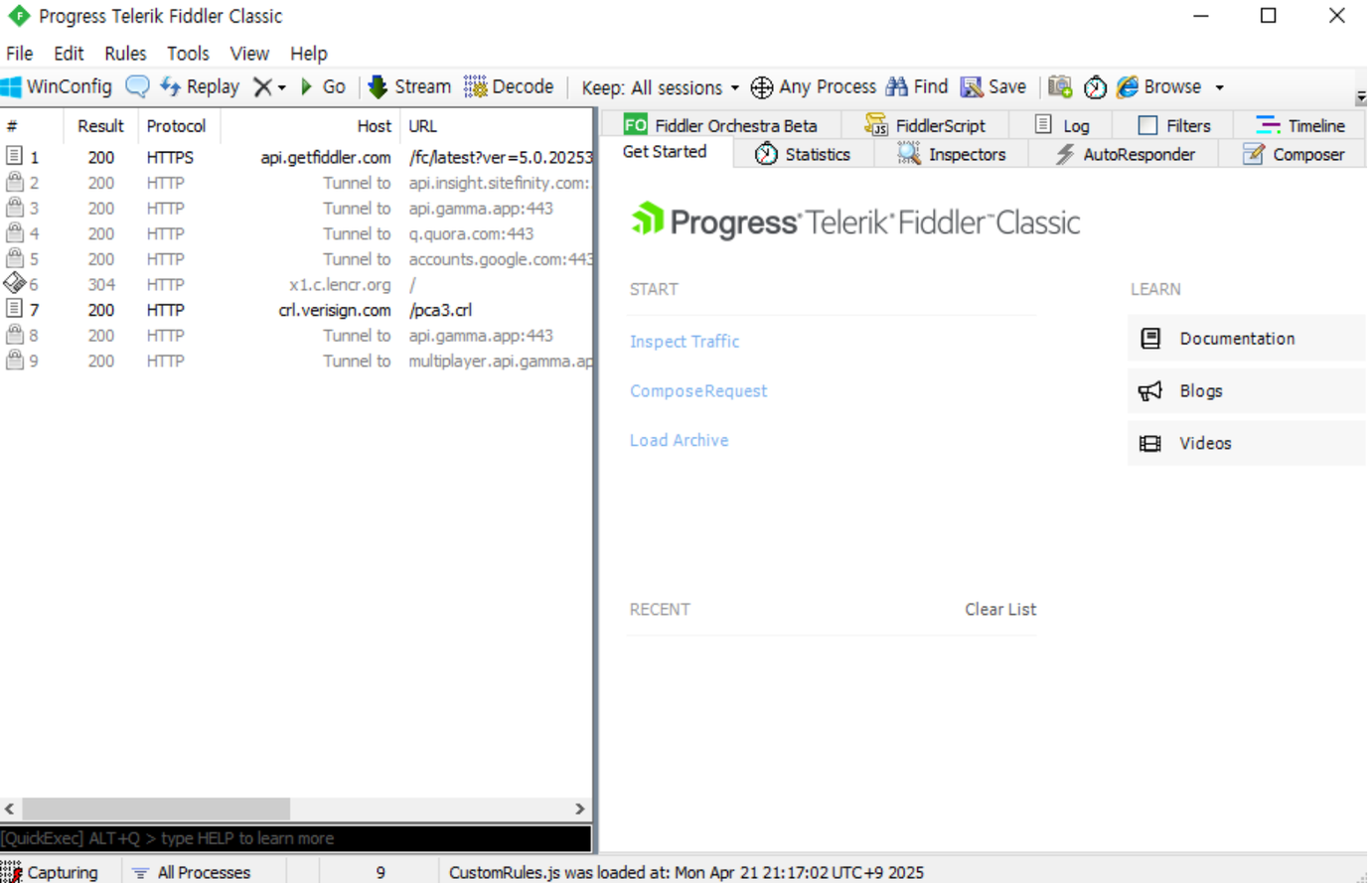
- Use the Fiddler Classic icon in your **START | Programs** menu.
- Or type *fiddler* in **Start | Run**

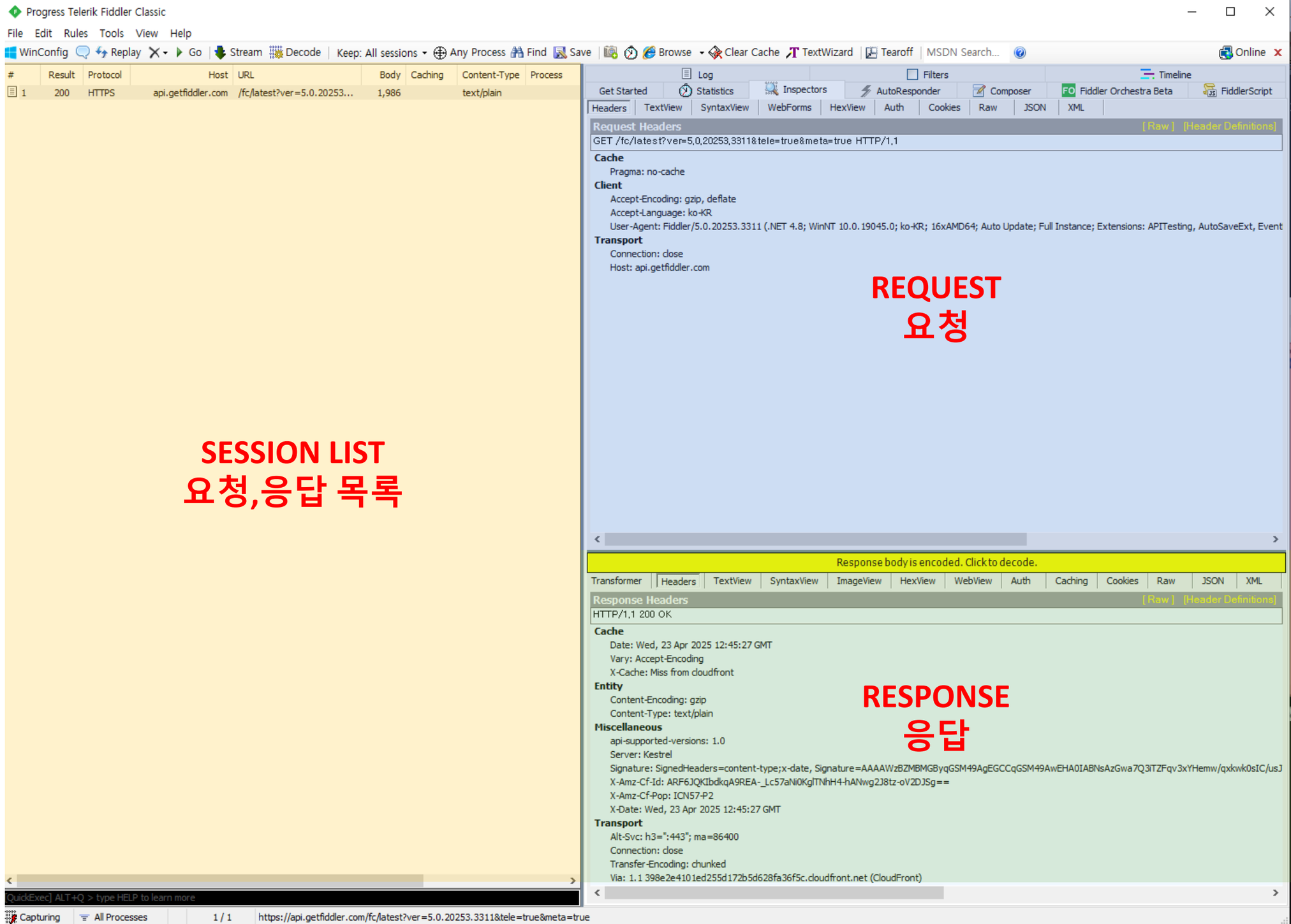
Important configuration steps

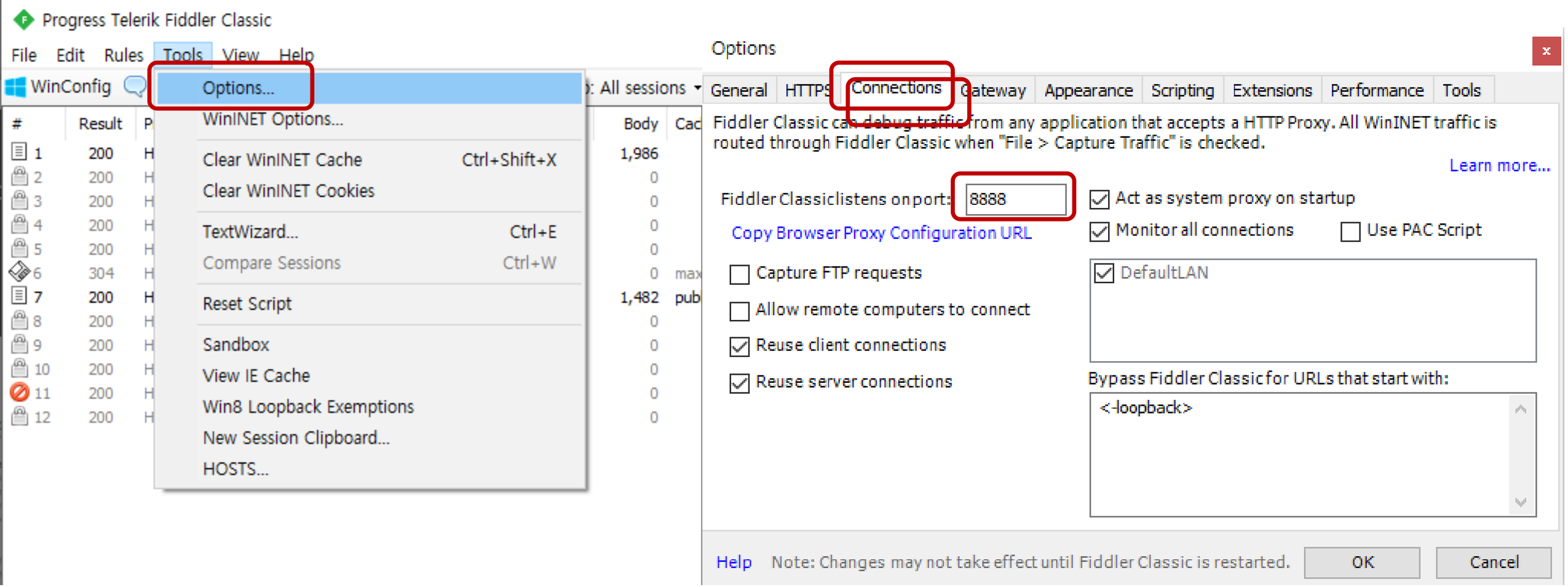
- Configure Fiddler Classic for Windows 8
- Configure Fiddler Classic to decrypt HTTPS traffic
- Monitor traffic to localhost from IE or .NET


최근에 추가한 앱
Fiddler ScriptEditor
Fiddler Classic


AppContainer Configuration
Windows uses an isolation technology called 'AppContainer' that may interfere with traffic capture from Immersive Applications and the Edge browser. Use the WinConfig button in Progress Telerik Fiddler Classic's toolbar to enable traffic capture.
Would you like to learn more?
To disable this warning, click 'Cancel'.
Yes
No
Cancel


Progress Telerik Fiddler Classic
File Edit Rules Tools View Help
WinConfig Replay Go Stream Decode Keep: All sessions Any Process Find Save Browse
Result Protocol Host URL
1 200 HTTPS api.getfiddler.com /fc/latest?ver=5.0.20253
2 200 HTTP Tunnel to api.insight.sitefinity.com:
3 200 HTTP Tunnel to api.gamma.app:443
4 200 HTTP Tunnel to q.quora.com:443
5 200 HTTP Tunnel to accounts.google.com:443
6 304 HTTP x1.c.lencr.org /
7 200 HTTP crl.verisign.com /pca3.crl
8 200 HTTP Tunnel to api.gamma.app:443
9 200 HTTP Tunnel to multiplayer.api.gamma.ap
[QuickExec] ALT+Q > type HELP to learn more
Capturing All Processes 9
CustomRules.js was loaded at: Mon Apr 21 21:17:02 UTC+9 2025
Fiddler Orchestra Beta FiddlerScript Log Filters Timeline
Get Started Statistics Inspectors AutoResponder Composer
Progress Telerik Fiddler Classic
START
Inspect Traffic
ComposeRequest
Load Archive
LEARN
Documentation
Blogs
Videos
RECENT
Clear List


Progress Telerik Fiddler Classic
File Edit Rules Tools View Help
WinConfig Replay Go Stream Decode Keep: All sessions Any Process Find Save Browse Clear Cache TextWizard Tearoff MSDN Search... Online
Result Protocol Host URL Body Caching Content-Type Process
1 200 HTTPS api.getfiddler.com /fc/latest?ver=5.0.20253... 1,986 text/plain
SESSION LIST
요청,응답 목록
Log Filters Timeline
Get Started Statistics Inspectors AutoResponder Composer Fiddler Orchestra Beta FiddlerScript
Headers TextView SyntaxView WebForms HexView Auth Cookies Raw JSON XML
Request Headers [Raw] [Header Definitions]
GET /fc/latest?ver=5,0,20253,3311&tele=true&meta=true HTTP/1,1
Cache
Pragma: no-cache
Client
Accept-Encoding: gzip, deflate
Accept-Language: ko-KR
User-Agent: Fiddler/5.0.20253.3311 (.NET 4.8; WinNT 10.0.19045.0; ko-KR; 16xAMD64; Auto Update; Full Instance; Extensions: APITesting, AutoSaveExt, Event
Transport
Connection: close
Host: api.getfiddler.com
REQUEST
요청
Response body is encoded. Click to decode.
Transformer Headers TextView SyntaxView ImageView HexView WebView Auth Caching Cookies Raw JSON XML
Response Headers [Raw] [Header Definitions]
HTTP/1,1 200 OK
Cache
Date: Wed, 23 Apr 2025 12:45:27 GMT
Vary: Accept-Encoding
X-Cache: Miss from cloudfront
Entity
Content-Encoding: gzip
Content-Type: text/plain
Miscellaneous
api-supported-versions: 1.0
Server: Kestrel
Signature: SignedHeaders=content-type;x-date, Signature=AAAAWzBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IABNsAzGwa7Q3iTZFqv3xYHemw/qxkwk0sIC/usJ
X-Amz-Cf-Id: ARF6JQKIbdkqA9REA-_Lc57aNi0KglTNhH4-hANwg2J8tz-oV2DJSg==
X-Amz-Cf-Pop: ICN57-P2
X-Date: Wed, 23 Apr 2025 12:45:27 GMT
Transport
Alt-Svc: h3=":443"; ma=86400
Connection: close
Transfer-Encoding: chunked
Via: 1.1 398e2e4101ed255d172b5d628fa36f5c.cloudfront.net (CloudFront)
RESPONSE
응답
QuickExec] ALT+Q > type HELP to learn more
Capturing All Processes 1 / 1 https://api.getfiddler.com/fc/latest?ver=5.0.20253.3311&tele=true&meta=true



Tool – Options- Connections – 8888포트 확인

프록시

자동 프록시 설정

이더넷 또는 Wi-Fi 연결에 프록시 서버를 사용합니다. 이 설정은 VPN 연결에 적용되지 않습니다.

자동으로 설정 검색

켬

설정 스크립트 사용

끔

스크립트 주소

저장

수동 프록시 설정

이더넷 또는 Wi-Fi 연결에 프록시 서버를 사용합니다. 이 설정은 VPN 연결에 적용되지 않습니다.

프록시 서버 사용

끔

주소 포트

다음 항목으로 시작하는 주소를 제외하고 프록시 서버를 사용합니다. 여러 항목은 세미콜론(;)으로 구분합니다.

localhost;127.0.0.1:21300;127.0.0.1:16107;

피들러 실행 전 후

프록시

자동 프록시 설정

이더넷 또는 Wi-Fi 연결에 프록시 서버를 사용합니다. 이 설정은 VPN 연결에 적용되지 않습니다.

자동으로 설정 검색

끔

설정 스크립트 사용

끔

스크립트 주소

저장

수동 프록시 설정

이더넷 또는 Wi-Fi 연결에 프록시 서버를 사용합니다. 이 설정은 VPN 연결에 적용되지 않습니다.

프록시 서버 사용

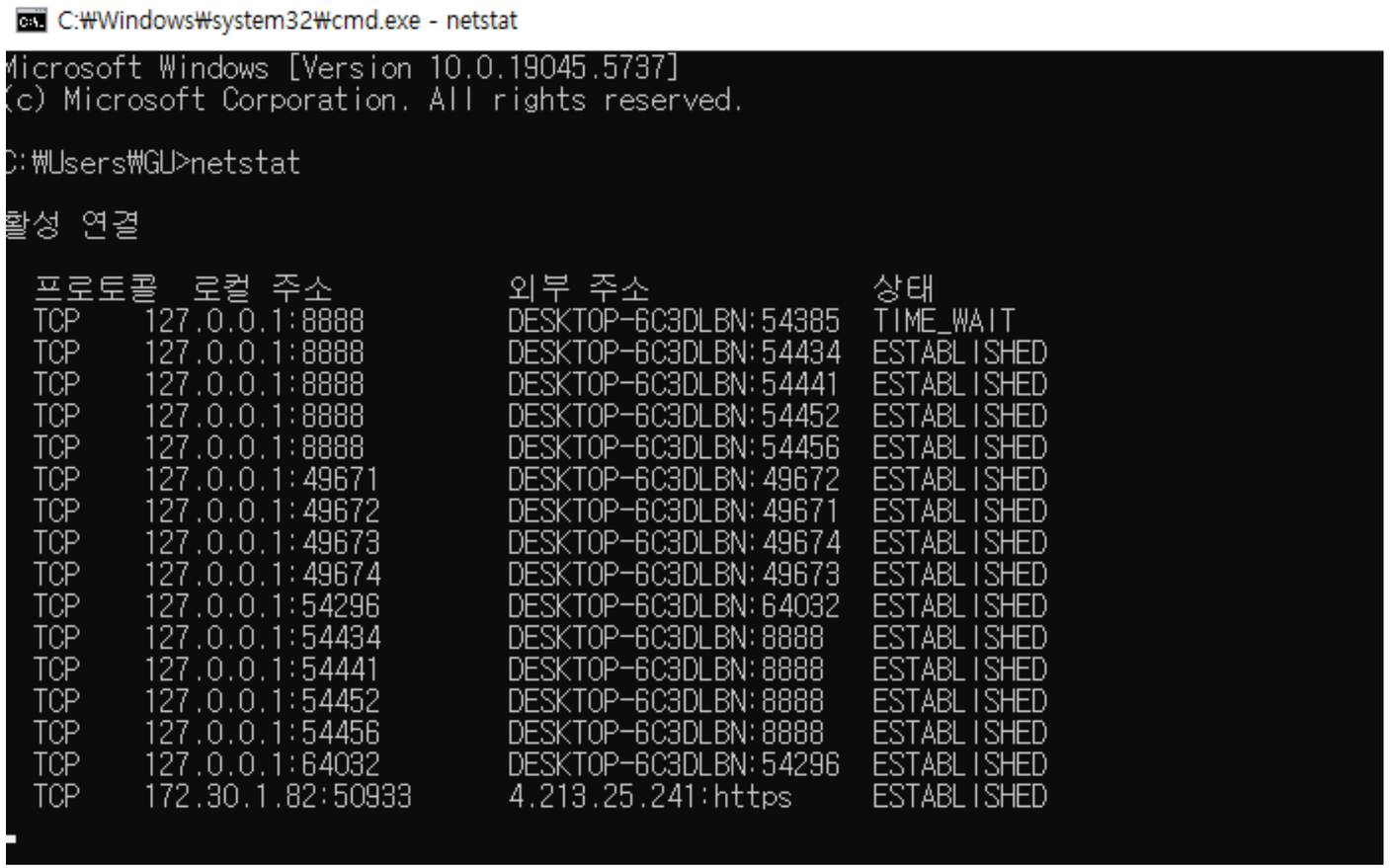
켬

주소 포트

http=127.0.0.1:8888;http:

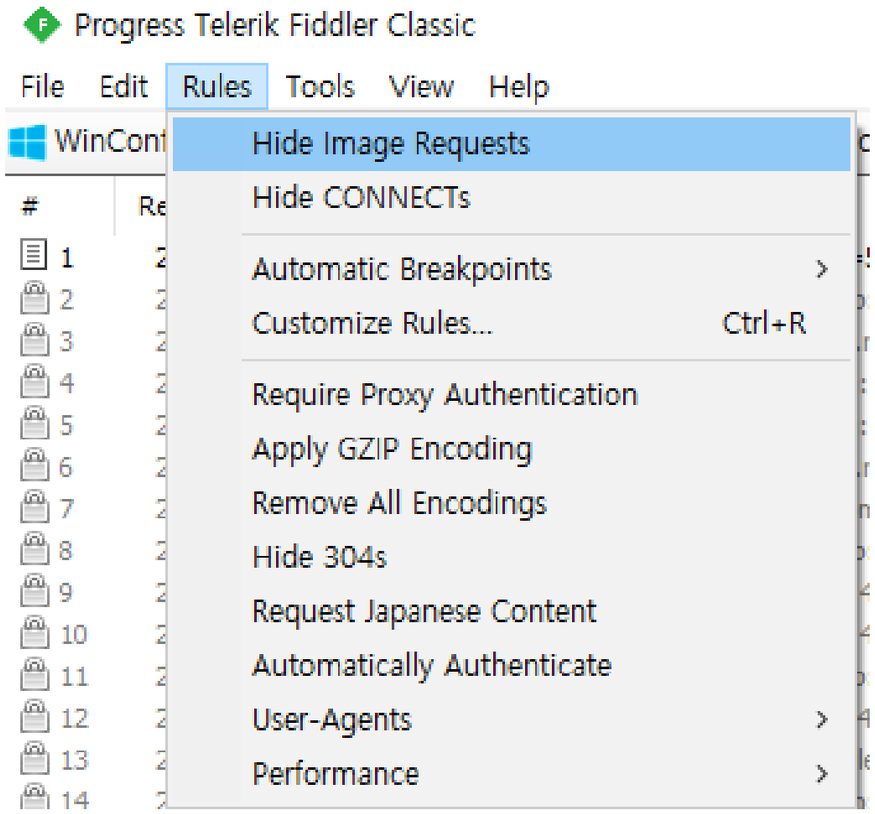
다음 항목으로 시작하는 주소를 제외하고 프록시 서버를 사용합니다. 여러 항목은 세미콜론(;)으로 구분합니다.

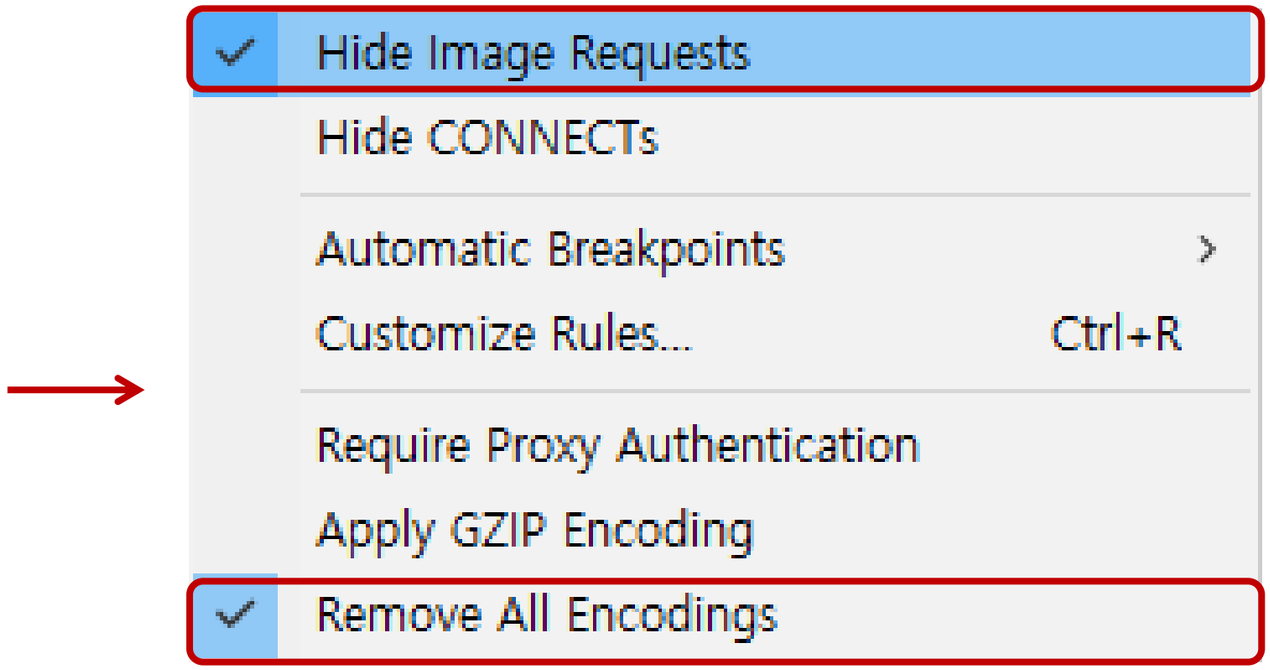
<-loopback>


C:₩Windows₩system32₩cmd.exe - netstat
Microsoft Windows [Version 10.0.19045.5737]
(c) Microsoft Corporation. All rights reserved.
C:₩Users₩GU>netstat
활성 연결
프로토콜 로컬 주소 외부 주소 상태
TCP 127.0.0.1:8888 DESKTOP-6C3DLBN:54385 TIME_WAIT
TCP 127.0.0.1:8888 DESKTOP-6C3DLBN:54434 ESTABLISHED
TCP 127.0.0.1:8888 DESKTOP-6C3DLBN:54441 ESTABLISHED
TCP 127.0.0.1:8888 DESKTOP-6C3DLBN:54452 ESTABLISHED
TCP 127.0.0.1:8888 DESKTOP-6C3DLBN:54456 ESTABLISHED
TCP 127.0.0.1:49671 DESKTOP-6C3DLBN:49672 ESTABLISHED
TCP 127.0.0.1:49672 DESKTOP-6C3DLBN:49671 ESTABLISHED
TCP 127.0.0.1:49673 DESKTOP-6C3DLBN:49674 ESTABLISHED
TCP 127.0.0.1:49674 DESKTOP-6C3DLBN:49673 ESTABLISHED
TCP 127.0.0.1:54296 DESKTOP-6C3DLBN:64032 ESTABLISHED
TCP 127.0.0.1:54434 DESKTOP-6C3DLBN:8888 ESTABLISHED
TCP 127.0.0.1:54441 DESKTOP-6C3DLBN:8888 ESTABLISHED
TCP 127.0.0.1:54452 DESKTOP-6C3DLBN:8888 ESTABLISHED
TCP 127.0.0.1:54456 DESKTOP-6C3DLBN:8888 ESTABLISHED
TCP 127.0.0.1:64032 DESKTOP-6C3DLBN:54296 ESTABLISHED
TCP 172.30.1.82:50933 4.213.25.241:https ESTABLISHED

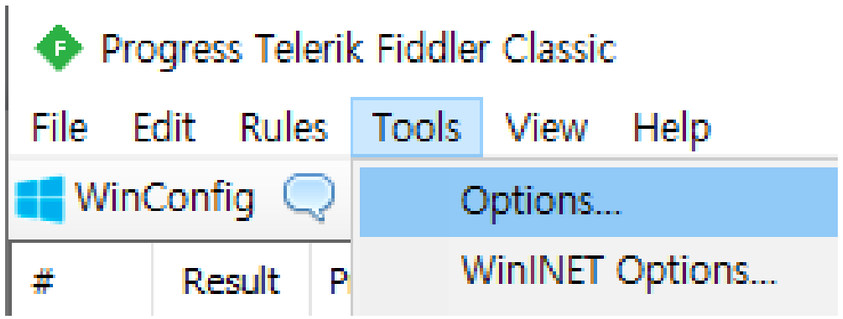
netstat

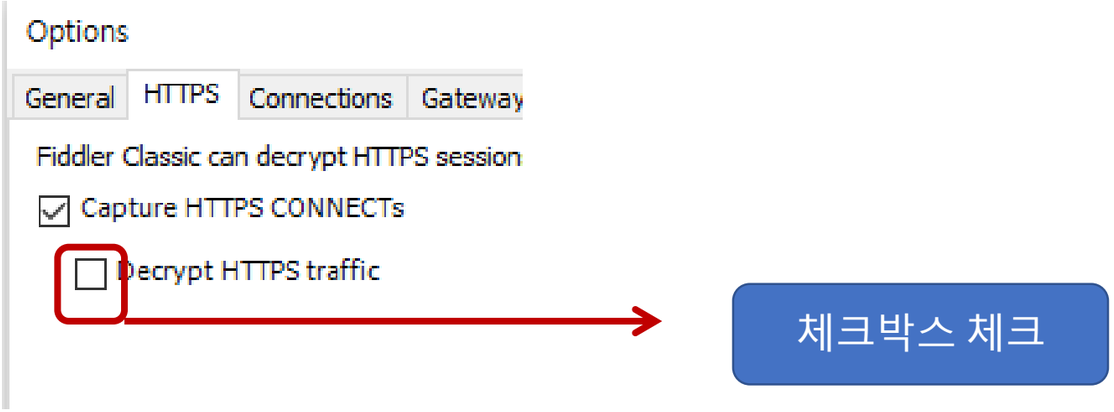
기본 옵션 변경


Progress Telerik Fiddler Classic
File
Edit
Rules
Tools
View
Help
Hide Image Requests
Hide CONNECTs
Automatic Breakpoints
Customize Rules... Ctrl+R
Require Proxy Authentication
Apply GZIP Encoding
Remove All Encodings
Hide 304s
Request Japanese Content
Automatically Authenticate
User-Agents
Performance

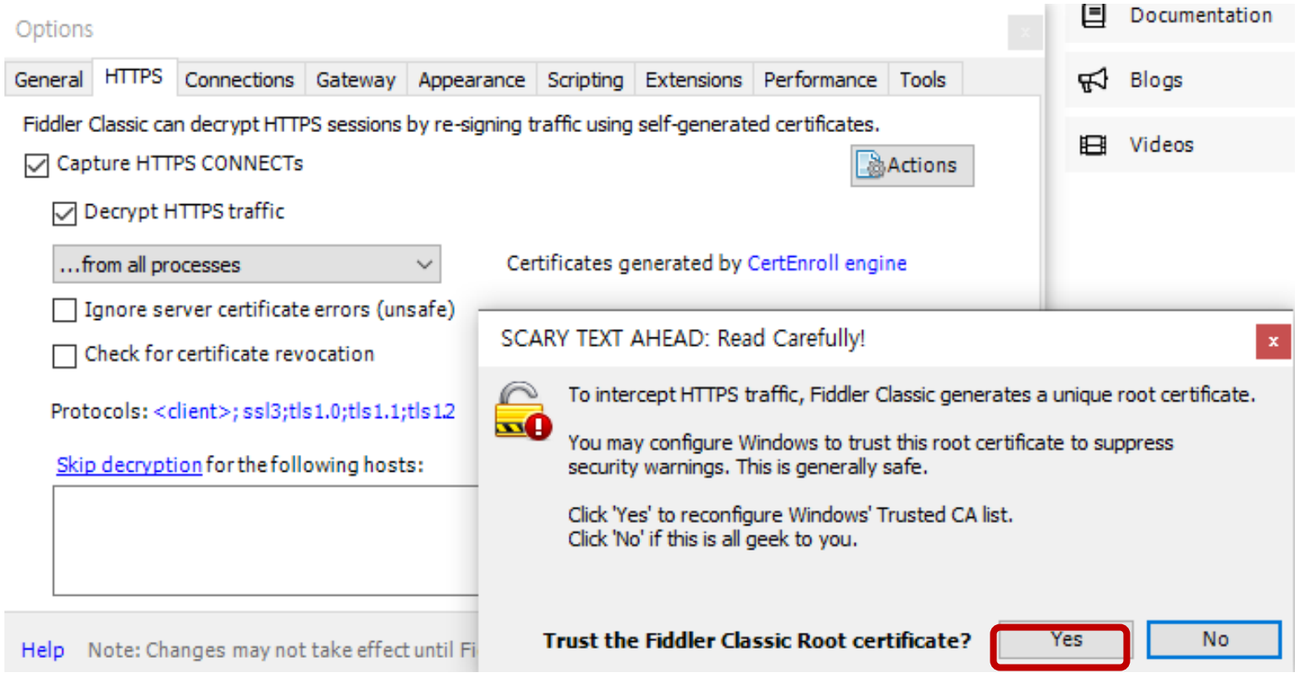

Hide Image Requests
Hide CONNECTs
Automatic Breakpoints
Customize Rules... Ctrl+R
Require Proxy Authentication
Apply GZIP Encoding
Remove All Encodings

SSL 세팅



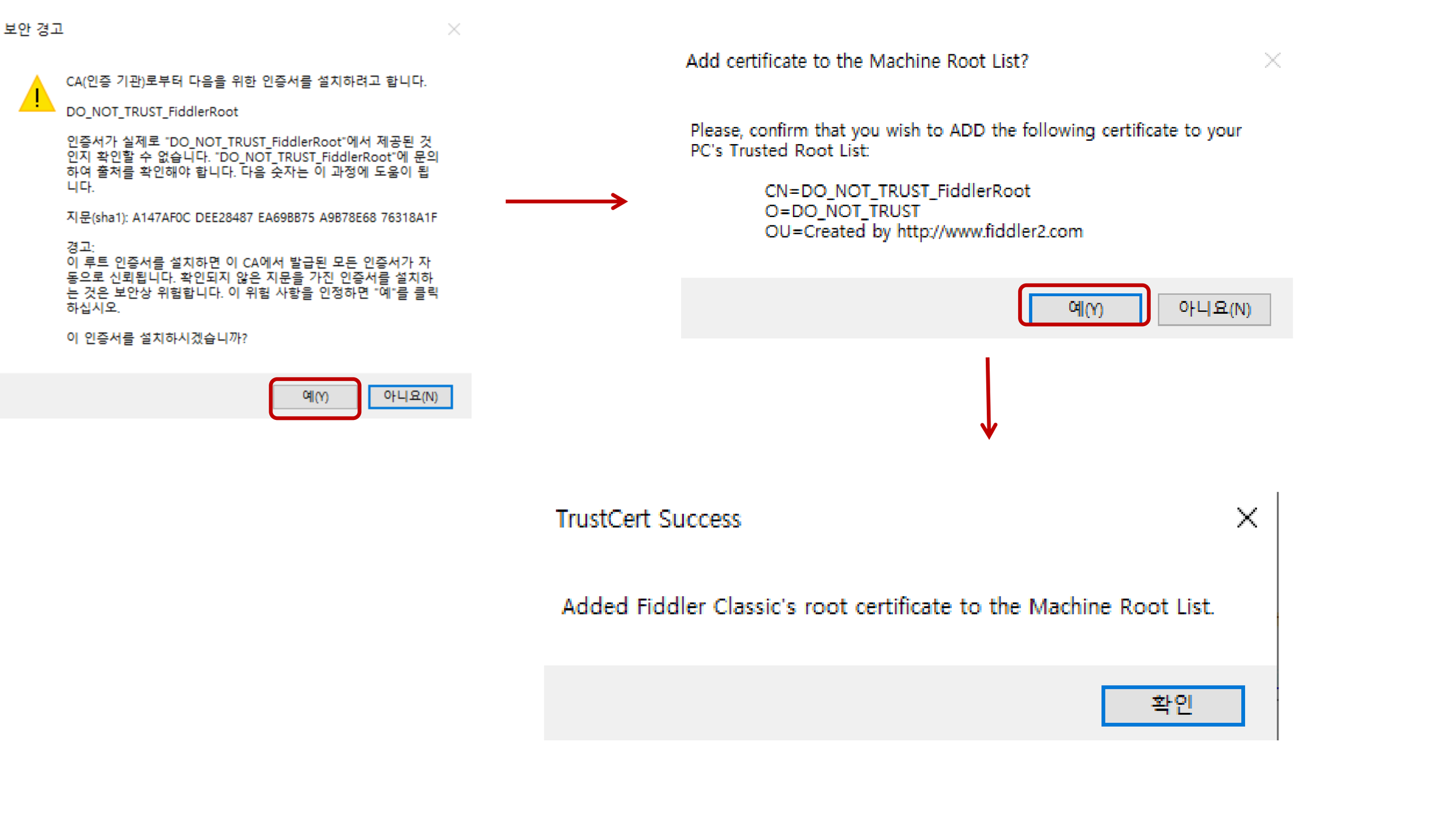


체크박스 체크



SSL(Secure Sockets Layer)

- 인터넷에서 데이터를 안전하게 주고받기 위한 암호화 통신 방식
- 웹 주소가 https://로 시작하는 모든 사이트는 SSL/TLS를 사용


보안 경고
CA(인증 기관)로부터 다음을 위한 인증서를 설치하려고 합니다.
DO_NOT_TRUST_FiddlerRoot
인증서가 실제로 "DO_NOT_TRUST_FiddlerRoot"에서 제공된 것인지 확인할 수 없습니다. "DO_NOT_TRUST_FiddlerRoot"에 문의하여 출처를 확인해야 합니다. 다음 숫자는 이 과정에 도움이 됩니다.
지문(sha1): A147AF0C DEE28487 EA69BB75 A9B78E68 76318A1F
경고:
이 루트 인증서를 설치하면 이 CA에서 발급된 모든 인증서가 자동으로 신뢰됩니다. 확인되지 않은 지문을 가진 인증서를 설치하는 것은 보안상 위험합니다. 이 위험 사항을 인정하면 "예"를 클릭하십시오.
이 인증서를 설치하시겠습니까?
예(Y)
아니요(N)
Add certificate to the Machine Root List?
Please, confirm that you wish to ADD the following certificate to your PC's Trusted Root List:
CN=DO_NOT_TRUST_FiddlerRoot
O=DO_NOT_TRUST
OU=Created by http://www.fiddler2.com
예(Y)
아니요(N)
TrustCert Success
Added Fiddler Classic's root certificate to the Machine Root List.
확인

1. 크롬 브라우저 실행

2.피들러에서 확인

클라이언트 코드 작성하기

- 코드 생성

피들러로 클라이언트 코드 작성하기

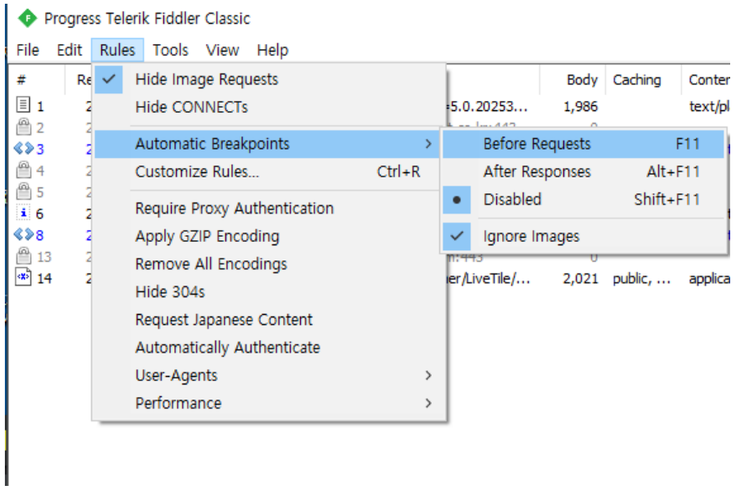
폼 값 변경

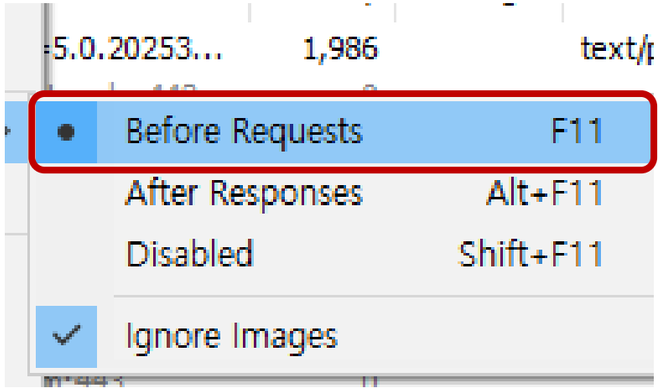
가격·수량·할인 등 입력값을 조작

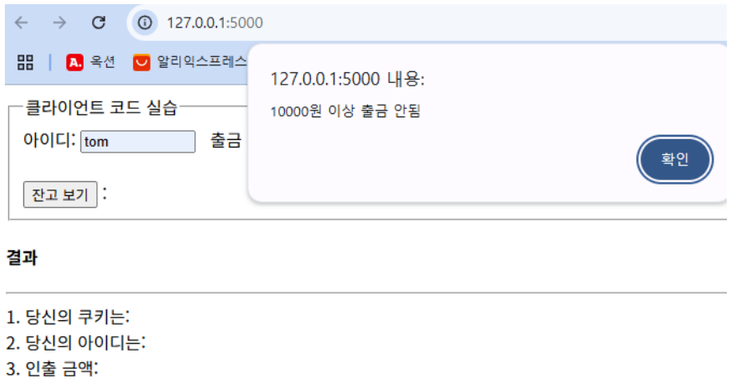
피들러로 클라이언트 코드 작성하기

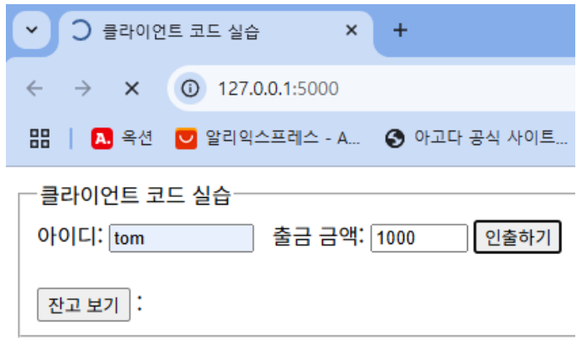
폼 값 변경

가격·수량·할인 등 입력값을 조작









rules – automatic Breakpoints – before Requests(F11) 체크

클라이언트 코드 조작

폼 값 변경

가격·수량·할인 등 입력값을 조작

DOM/JS 수정

입력 제한 해제와 숨김 정보 노출

자바스크립트 삭제 및 변조

클라이언트 방어 로직 무력화

쿠키/세션 위조

인증 정보 탈취, 권한 변경 시도

실습 도구: **Fiddler, Burp Suite**

옵션 이름	기능 요약	의미
Before Requests (F11)	요청(Request)을 서버에 보내기 전에 Fiddler가 가로채서 멈춤	클라이언트 → 서버 방향 요청을 **보내기 전에 중단**
After Response (Alt+F11)	서버에 요청(Request)을 보내고 나서 Fiddler가 멈춤	요청을 서버로 보낸 **직후에 중단**

조작 실습 예시 (1) – 폼 값 변경

1 폼 제한 확인

JS로 인출 제한 설정

2 Fiddler/프록시 사용

요청 데이터 조작, 제한 우회

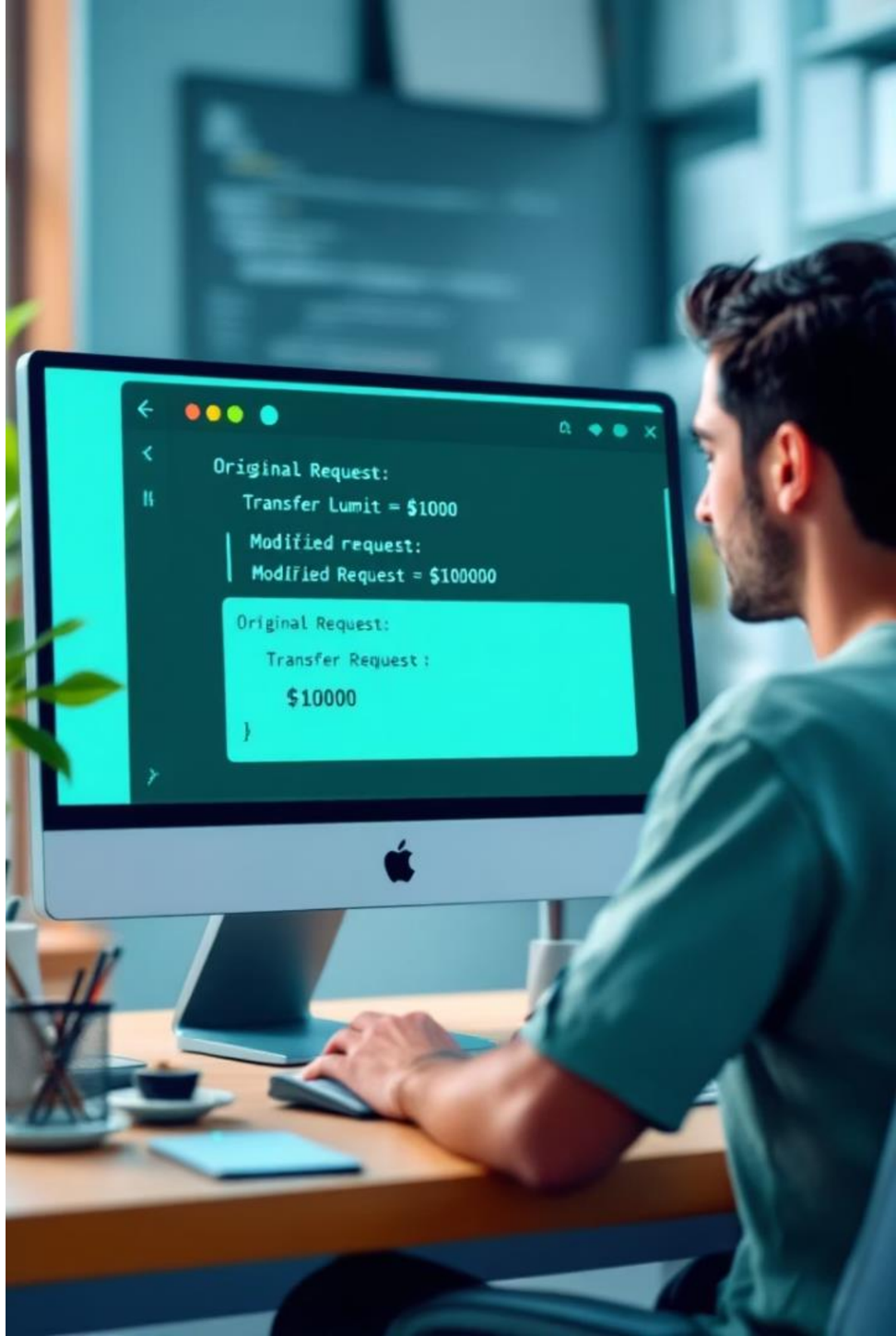
3 서버 미검증 위험

서버 검증 없으면 무제한 인출

4 대응 방안

모든 입력값 서버에서 재확인

- rules – automatic Breakpoints – before Requests(F11) 체크



조작 실습 예시
(2) – DOM & JS

maxlength 제한 우회

입력 길이 제한 무시, ID 등 입력

자바스크립트 조건 삭제

의도된 방어로직 무력화

UI는 환상에 불과

진짜 검증은 서버에서만 의미

- rules – automatic Breakpoints – after Requests(Alt + F11) 체크

조작 실습 예시
(2) – DOM & JS

maxlength 제한 우회

입력 길이 제한 무시, ID 등 입력

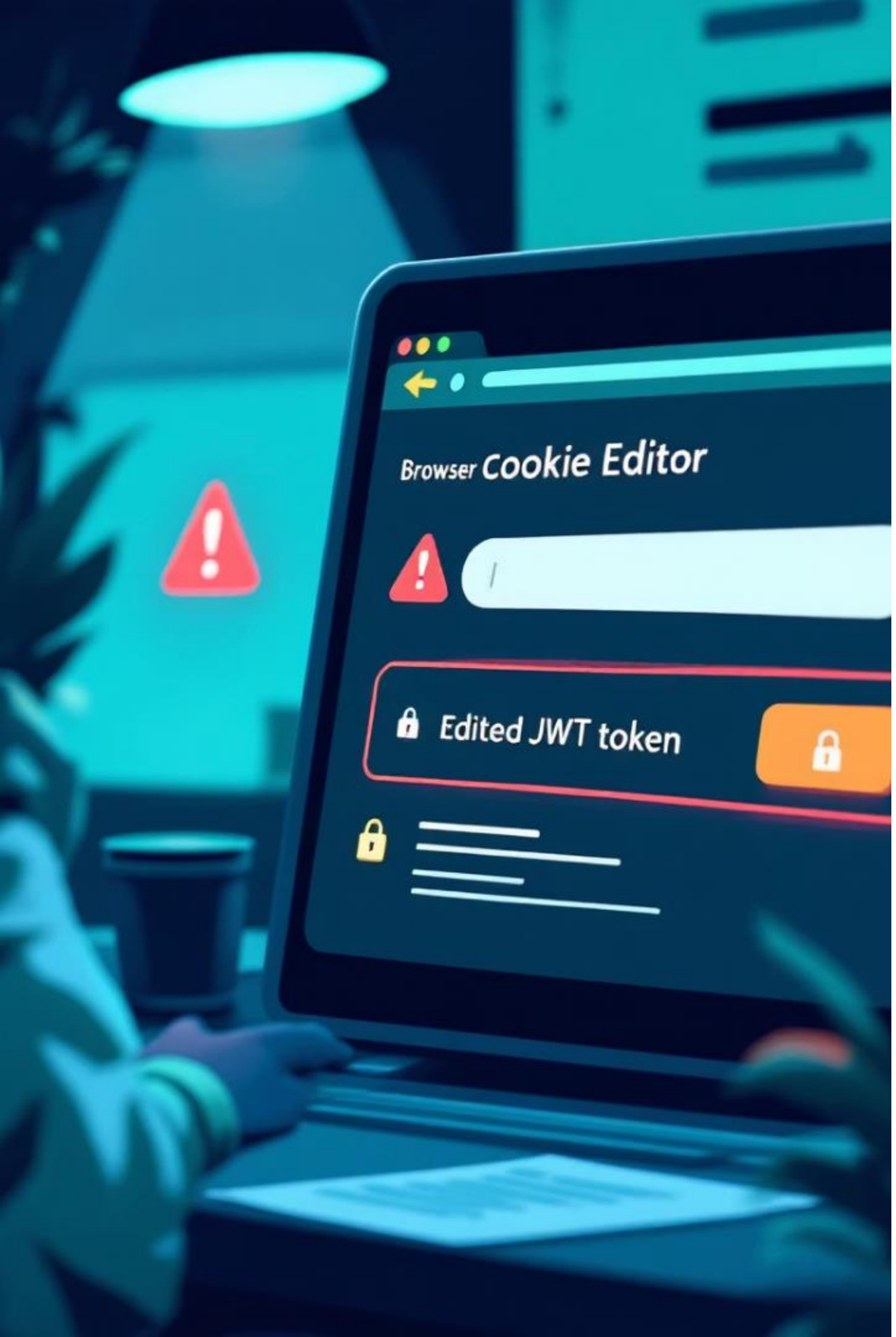
자바스크립트 조건 삭제

의도된 방어로직 무력화

UI는 환상에 불과

진짜 검증은 서버에서만 의미

- rules – automatic Breakpoints – after Requests(Alt + F11) 체크



조작 실습 예시 (3) – 쿠키

쿠키 조작

사용자 권한 직접 변경

암호화만의 한계

암호화되어도 조작은 가능

토큰 재생 공격

JWT 등도 동일하게 취약

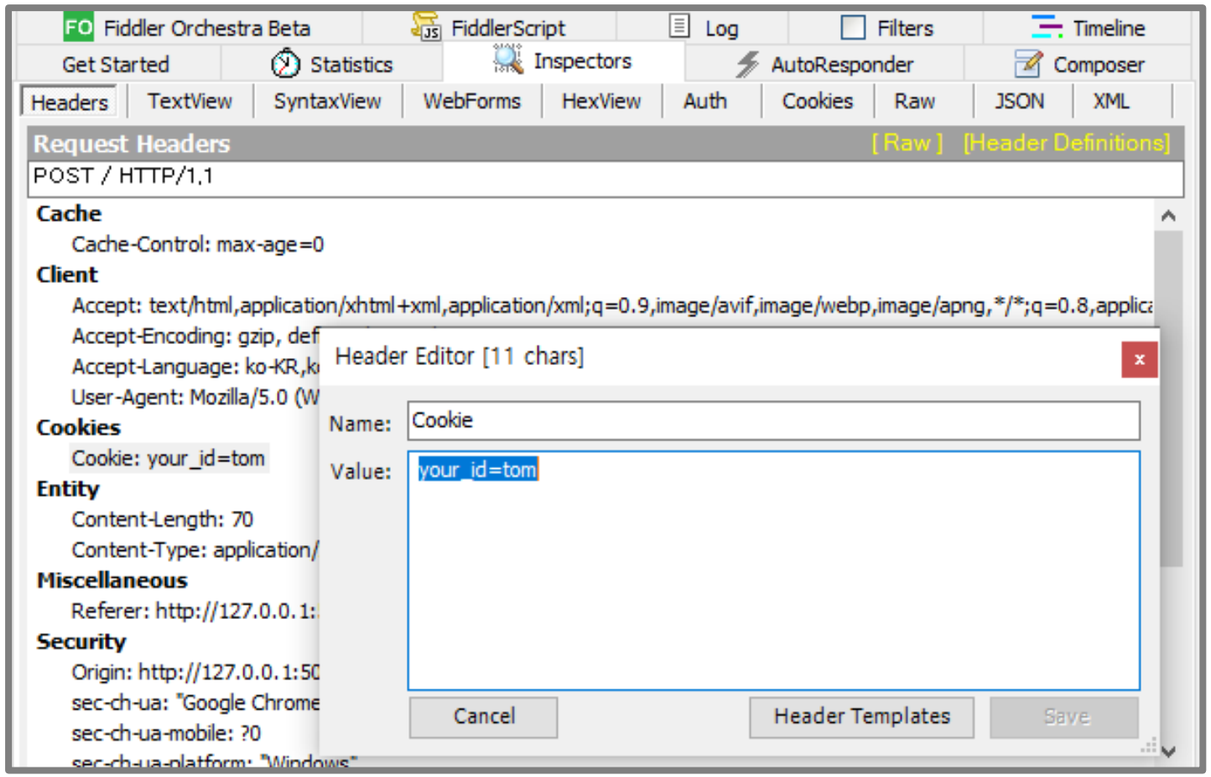


그림1

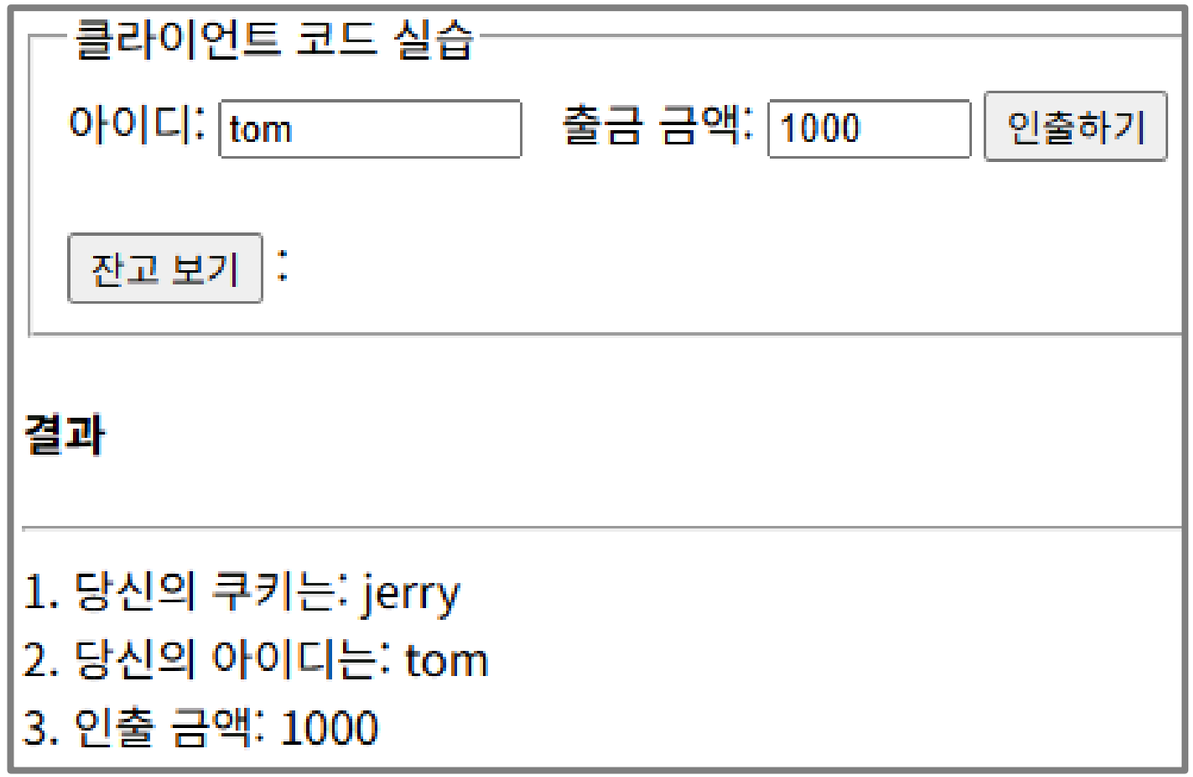


그림2

그림1. 피들러 가서 request -> Header 탭 선택 -> cookies 마우스 오른쪽 클릭

그림2. edit header 클릭 -> jerry로 수정후 저장버튼 클릭 -> 브라우저에서 쿠키값이 바뀜

AJAX & API 데이터 조작

- **AJAX로 데이터 송수신**

  JSON/Text 구조 노출

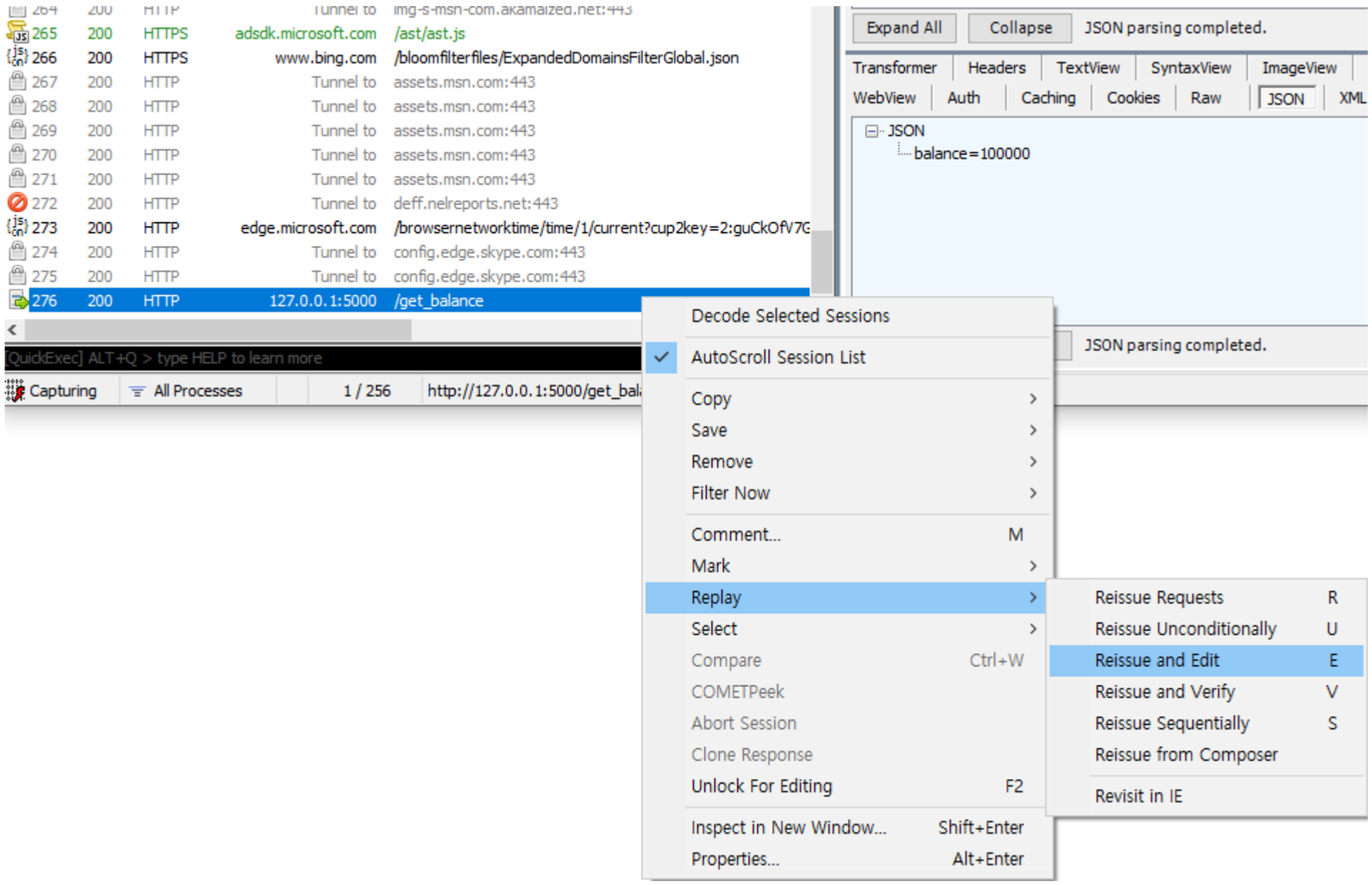
  - **텍스트 기반 요청 변조**

    권한 없는 데이터 접근 시도

    - **비동기 호출 안전 착각**

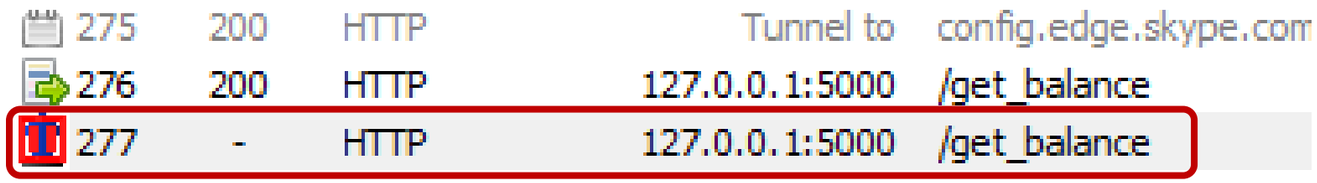
      실제론 쉽게 변조 가능

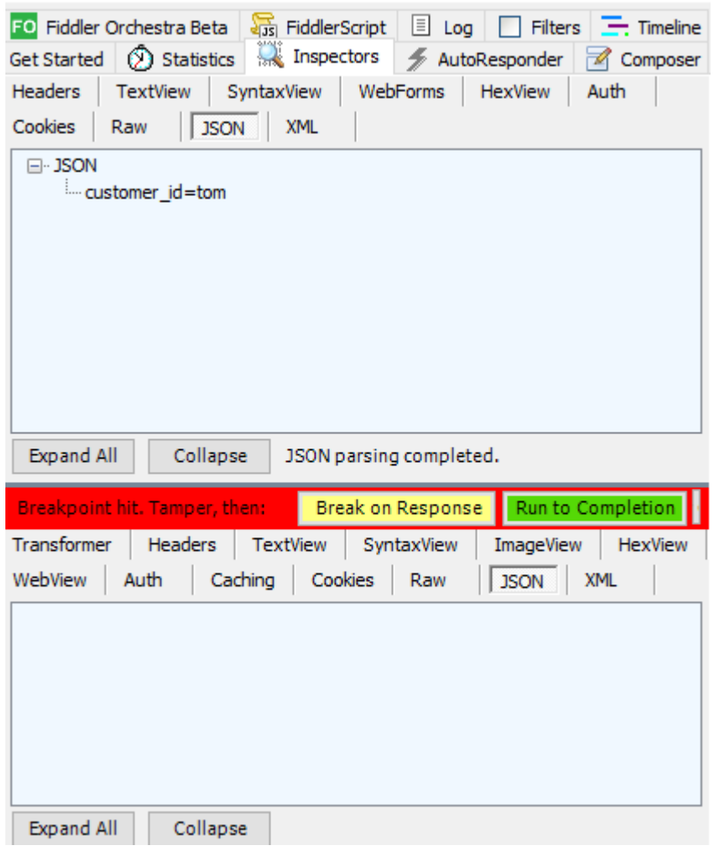
Replay

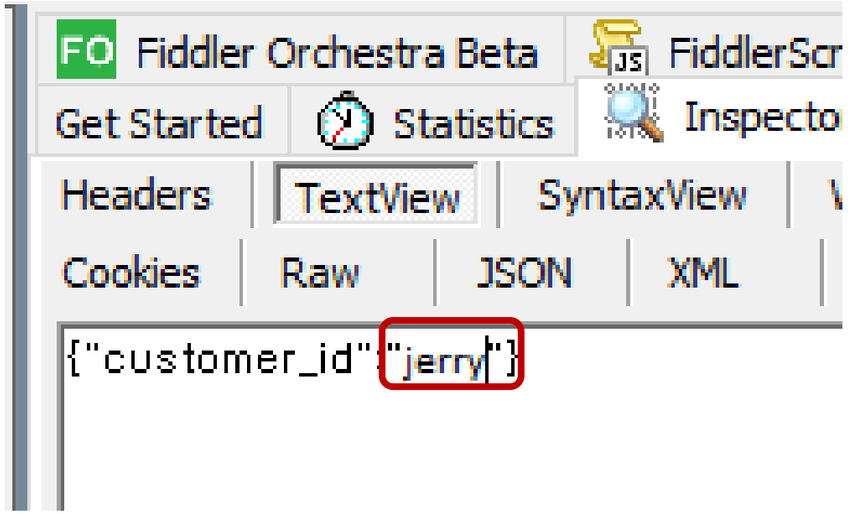


1. url 세션에서 /get_balance 선택 -> 마우스 오른쪽 -> Replay -> Reissue and Edit 선택

Replay

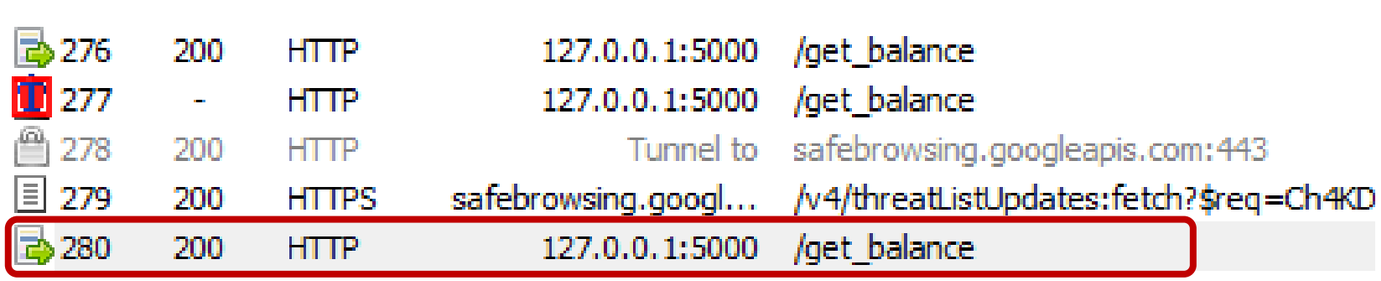


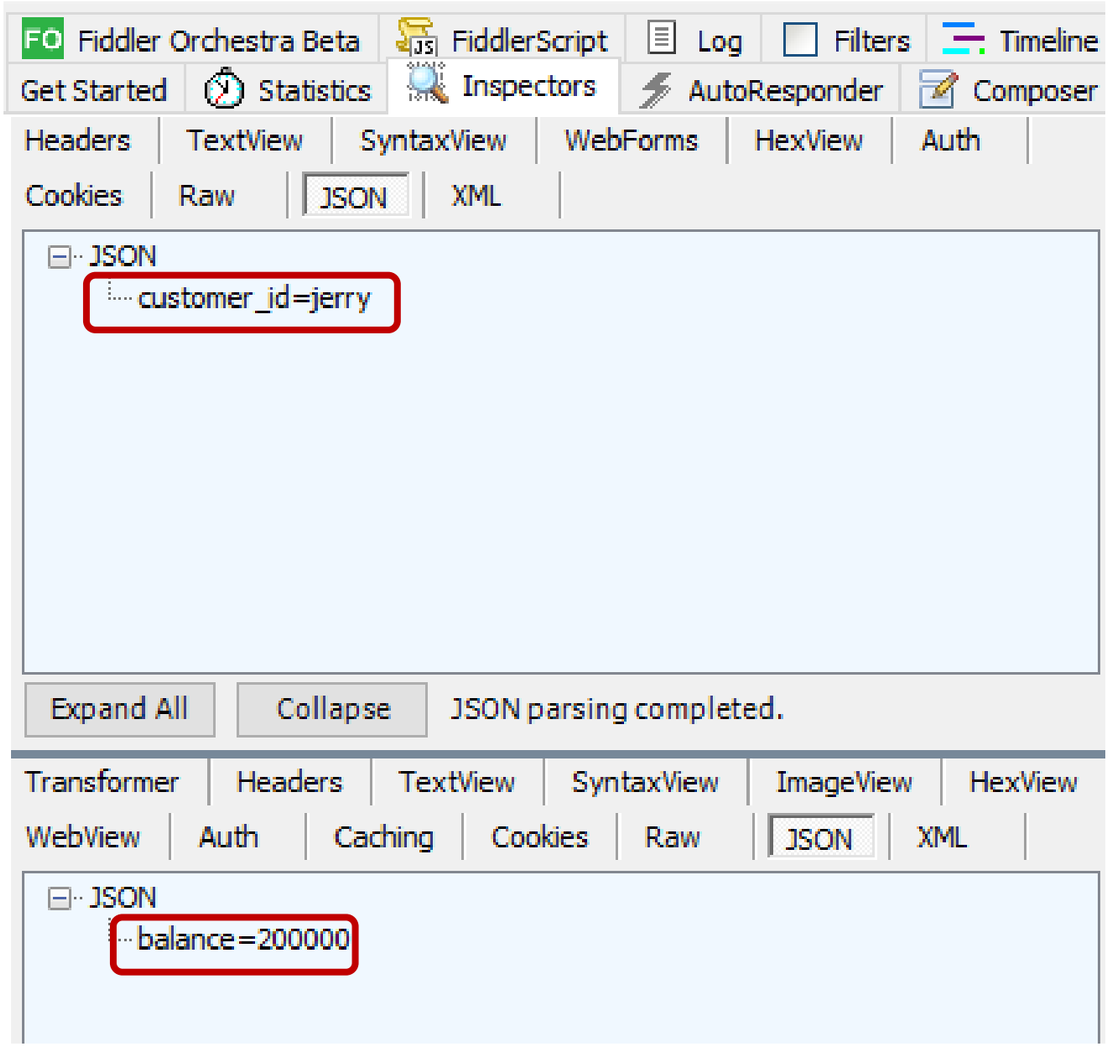




새로운 /get_balance 호출 생성

Request -> Text View탭 -> jerry로 수정
Run to Completion 버튼 실행
수정한 json 값으로 url 호출





호출된 url 클릭

Request -> json

- custom_id = jerry

Response -> json

- balance = 200000

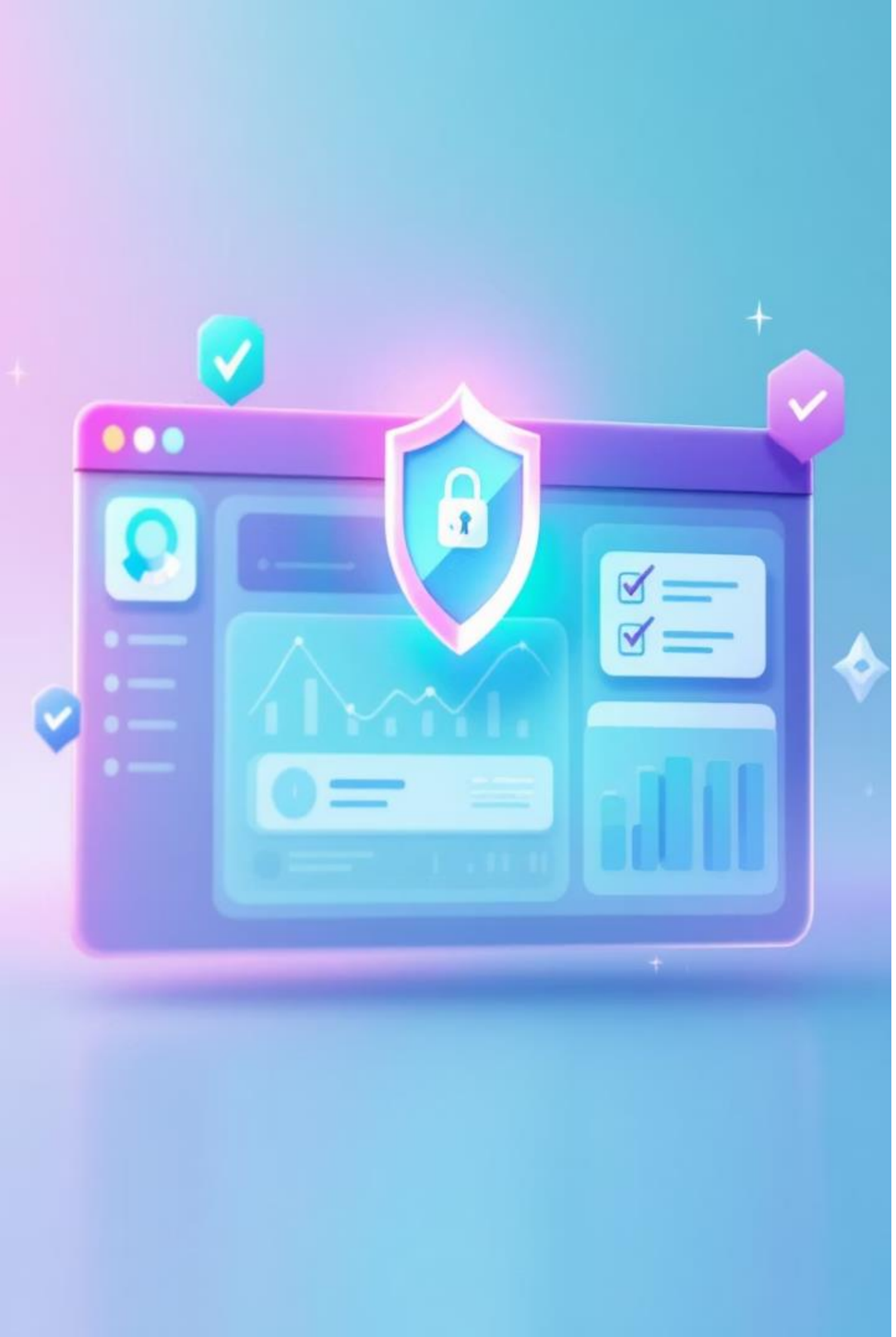
웹프레임 워크 MVC 패턴

- **Model-View-Controller**의 약자
- 프로그램을 **3가지 역할로 분리**해서 만들자는 구조

역할	설명
Model	데이터와 비즈니스 로직을 관리 (DB, 데이터 저장, 처리)
View	사용자에게 보여지는 화면 (HTML, 화면 UI)
Controller	Model과 View를 연결하고 사용자 입력을 처리 (요청 처리, 흐름 제어)

과거 웹 프레임워크 vs 2025년 웹 프레임워크 비교

항목	과거 웹 프레임워크 (2000~2010년대 초)	최신 웹 프레임워크 (2025년 현재)
개발 패턴	전통적 MVC 또는 직접 템플릿 조합	진보된 MVC + API 중심 개발 (Headless)
주된 통신	주로 전체 HTML을 통째로 요청/응답 (Full Page Reload)	JSON 기반 비동기 통신 (API 호출, SPA)
예시	HP(Laravel), Ruby on Rails, JSP(Spring MVC)	FastAPI, Next.js, Nuxt.js, NestJS
View 처리	서버가 HTML을 생성해서 클라이언트로 전송	프론트엔드(Vue, React, Svelte)가 View 렌더링, 서버는 데이터(API)만 전달
Controller 역할	요청을 받아서 직접 HTML 뷰를 연결	요청을 받아 데이터(API)로 응답. View는 별도 프론트엔드가 담당
Model 역할	ORM 또는 SQL 직접 연결 (DB관리)	ORM + Microservice API 통합 (DB + 외부 서비스 연동)
대표적 특징	서버 중심(Server-Side Rendering)	클라이언트 중심(Client-Side Rendering) + 서버 API만 담당
주요 키워드	JSP, 서블릿, PHP, MVC Framework	REST API, GraphQL, Headless CMS, Serverless Architecture



요약 및 안전한 웹을 위한 전략

- 클라이언트 코드 신뢰 금지
- 서버사이드 최종 검증 로직 필수
- API 권한 체크 + 암호화 + 만료 정책
- 보안 설계 시점부터 클라이언트 동선 고려
- OWASP Top 10, Mobile Top 10과 연결해 점검